NewMsg-Fly8000 无线开发套件使用手册

V2.0

杭州威步科技有限公司

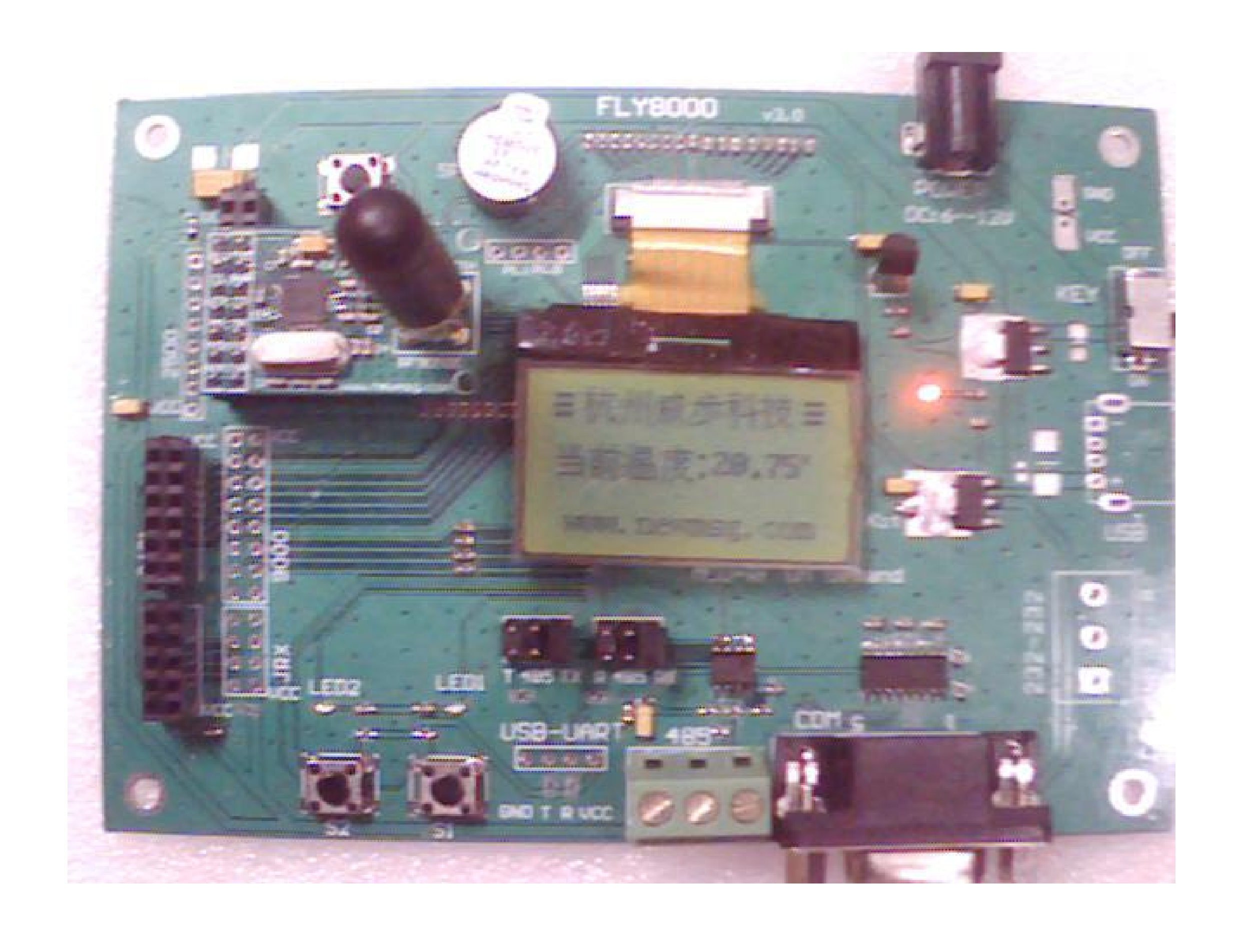
2009 - 4 - 2

电话: 0571-81310832 13282105900 传真: 0571-86919079
网址: http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址: 浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

目录

电话： 0571-81310832 13282105900 **传真：** 0571-86919079
网址： http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼
416

第一章：Fly8000 开发板简介



1.1 为什么 Fly8000 选择 STC 系列 51 单片机

Fly8000 套件主控 MCU 采用 STC10F08XE，是宏晶新款 51 单片机，，其程序代码与传统的 STC89C52 一样，只是 STC10F08XE 多了 P4 口，只要带头文件中加上“sfr P4=0XC0”,做单片机开发的新、老工程师，对 51 单片机也是再熟悉不过了，因此就能快

速的开发项目。 STC10F08XE 具有 8K flash，512 字节 SRAM，内置 5K EEPROM 并且具有掉电唤醒功能。

Fly8000 还支持本公司大多数无线射频模块，比如 RF905、RF27A、RF30A、CC1100、RF2401、RF2500 等各种模块，能非常方便的应用于各种无线领域的开发。

1.2 Fly8000 硬件简介

1、经典的电源管理（11175.0、11173.3 为本套件稳定的输送电能）

2、串口 ISP，实现代码下载以及串口通信

3、各类无限射频接口

4、蜂鸣器，方便做音乐实验以及报警

5、两路按键以及两路 LED

6、LCD128*64 图形灰度液晶，能方便的绘制图形

7、DS18B20 方便检测外部环境温度

8、485 总线接口，方便进行 485 实验

9、引出电脑 232 电平接口

10、引出串口 TTL 电平，可配套本公司 USB 下载线进行 USB 下载

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN：wenming_hu2002@live.cn QQ：4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

11、外扩插针 P1.0、P1.1，兼容 STC PIN44 引脚 QFP 封装的所有单片机

12、USB 供电接口

1.3、使用说明

1、系统上电，注意电源电压，上电后，开发板上电源指示灯会点亮，表示供电正常。

2、本开发板下载代码时候，是通过串口，并且重新启动单片机实现下载，下载软件可到宏晶官方网站下载，最新版，也可以直接向本公司索取。

3、本套件能使用电池盒供电测试，在电源处有焊盘和插针，标准 DIP，方便移动测试距离等用途。

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

第二章 典型无线模块简介

2.1 CC1100 无线模块

2.1.1 芯片特点及应用领域

CC1100 芯片特性（功能丰富，灵活性大，强大 WOR 功能）

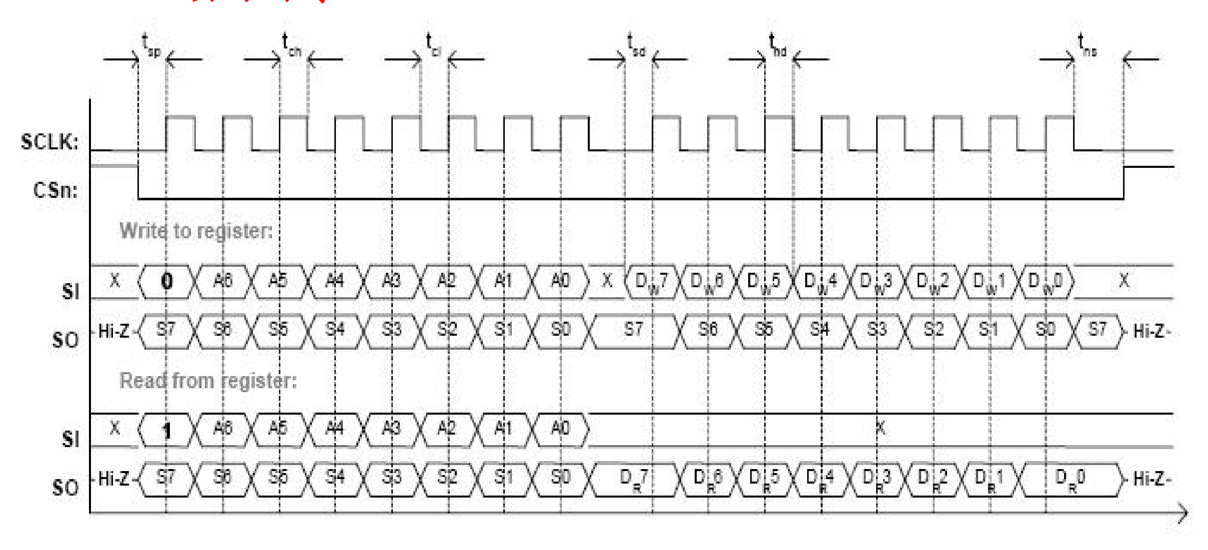
(1) 315、433、868、915Mh 的 ISM 和 SRD 频段
(2)最高工作速率 500kbps，支持 2-FSK、GFSK 和 MSK 调制方式选用外置 433 天线，空旷通讯距离可达 300 米左右，降低通信波特率距离更远，我公司也提供高精度参数 RF1100SE 模块，性能更佳，室内通信仍有良好通信效果，3-6 层可实现可靠通信，抗干扰性能强，很强的扰障碍穿透性能;
(3)高灵敏度（1.2kbps 下-110dDm，1％数据包误码率）
(4)内置硬件 CRC 检错和点对多点通信地址控制
(5)较低的电流消耗（RX 中，15.6mA，2.4kbps,433MHz）
(6)可编程控制的输出功率，对所有的支持频率可达+10dBm
(7)支持低功率电磁波激活功能，支持传输前自动清理信道访问（CCA），即载波侦听系统
(8)模块可软件设地址，软件编程非常方便
(9)标准 DIP 间距接口，便于嵌入式应用
(10)单独的 64 字节 RX和TX数据 FIFO应用领域:用于极低功率 UHF 无线收发器，315/433/868和915MHzISM/SRD 波段系统，自动仪表读数，电子消费产品，远程遥控控制，低功率遥感勘测，住宅和建筑自动控制，无线警报

和安全系统，工业监测和控制，无线传感器网络,无线唤醒功能，低功耗手持终端产品。

2.1.2 CC1100 无线通信流程解析

单片无线收发器工作在433/868/915MHZ 的ISM 频段由一个完全集成的频率调制器一个带解调器的接收器一个功率放大器一个晶体震荡器和一个调节器组成。工作特点是自动产生前导码和 CRC可以很容易通过SPI 接口进行编程配置，电流消耗低。

SPI 读写时序



2.2 RF905 无线模块

2.2.1 RF905 芯片特性以及应用

1）接收发送功能合一，收发完成中断标志

2）433/868/915 工作频段，433MHZ 开放 ISM 频段免许可使用

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

3）发射速率 5 0Kbps，选用外置 433 天线，空旷通讯距离可达 300米左右，加功放可到 3000 米左右；室内通信仍有良好通信效果，3-6
层可实现可靠通信，抗干扰性能强，很强的扰障碍穿透性能；
4）每次最多可发送接收 32 字节，并可软件设置发送/接收缓冲区大小 1/2/4/8/16/32 字节；
5）170 个频道，可满足多点通讯和跳频通讯需求，实现组网通讯， TDMA，CDMA，FDMA；
6）内置硬件 8/16 位 CRC 校验，开发更简单，数据传输可靠稳定。
7）1.9-3.6V 工作，低功耗，待机模式仅 2.5uA.
8） 内置 SPI 接口，也可通过 I/O 口模拟 SPI 实现。最高 SPI 时钟可达 10M。

应用领域: 车辆监控、遥控、遥测、小型无线网络、无线抄表、门禁系统、小区传呼、工业数据采集系统、无线标签、身份识别、非接触RF 智能卡、小型无线数据终端、安全防火系统、无线遥控系统、生物信号采集、水文气象监控、机器人控制、无线 232 数据通信、无线485/422 数据通信等。

2.2.2 RF905 无线通信流程解析

PWR_UP	TRX_CE	TX_EN	工作模式
0	X	X	掉电和SPI编程模式
1	0	X	Standby 和SPI编程模式

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

1	1	0	ShockBurst RX模式
1	1	1	ShockBurst TX模式

发送端工作流程

RFC-30H进入ShockBurst 工作模式后，无线系统自动上电，加前导码和CRC 校验完成数据包合成。数据包以100kbs，GFSK，曼切斯特编码方式发送。ShockBurst 工作模式确保一个传输包发送开始后，总能完成，不管在发送过程中TRX_CE、TX_EN 如何被设置。当发送结束后，新的模式被激活。

(一). 系统初始化:

1. 系统初始化;
2. RFC-30H初始化，写入寄存器配置字
3. 写入接收地址和待发送数据包TxAddress[4]，TxRxBuf[TxRxBuf_Len]

(二). 发送数据:

1. TX_EN=1,PWR_UP=1, TRX_CE =0 并按照ShockBurst TX 模式时序进行编程;
2. SPI写PAYLAOD+ADDR ;
3. 将TRX_CE 电平置高，延时650us 激活RFC-30H切换到ShockBurst 传输（ShockBurst 确保一个传输包发送开始后，总能完成）;

4. 将TRX_CE 电平置低（结束数据传输），延时650us 使RFC-30H切换到Standby 模式 ；然后循环进行第二步。

接收端工作流程:

RFC-30H接收到有效数据包（CRC 校验正确）时，去掉前导码，地址和CRC 位，数据准备就绪，此时可通过MCU 以最高10MHZ速率进行SPI 读接收数据，可采用I/O模拟SPI 接口方式。

(一). 系统初始化:

1. RFC-30H 初始化: 写入寄存器配置字
2. 写入接收地址和待发送数据包，TxAddress[4]，TxRxBuf[TxRxBuf_Len]

(二). 接收数据

1. TX_EN=0, PWR_UP=1 ； //Standby 模式
2. 将TRX_CE 置高电平，延时650us 使RFC-30H切换到ShockBurst RX 模式;
3. 按照 ShockBurst RX 模式时序进行编程;
4. 检测CD，如果CD=1，则说明有相同载波存在，如果没有等待。
5. 当接收到有效地址后，AM 被置高。
6. 当收到有效数据包（CRC 校验正确）时，RFC-30H去掉前导码，地址位和CRC位，数据准备就绪时DR被置位。
7. 将TRX_CE 电平置低，延时650us 使RFC-30H切换到Standby 模式
8. MCU通过SPI 接口读出有效数据后，AM 和DR被置低
9. 循环进行第二步。

2.3 RF24L01

2.3.1 RF24L01 芯片特性

1）接收发送功能合一，收发完成中断标志
2）工作频率 2.4~2.524GHZ，2.4GHZ 开放 ISM 频段免许可使用
3）发射速率最高可达 2Mbps，内置 PCB 板载天线，空旷通讯距离可达 100 米左右，室内在 50 米左右；抗干扰性能强。
4）每次最多可发送接收 32 字节，并可软件设置发送/接收缓冲区大小 1/2/4/8/16/32 字节；
5）125 个频道，可满足多点通讯和跳频通讯需求，实现组网通讯，TDMA，CDMA，FDMA；
6）内置硬件 8/16 位 CRC 校验，开发更简单，数据传输可靠稳定。
7）在 2Mbit/s 速率下@0dBm 输出时的峰值电流 11mA，掉电模式下的功耗 400nA，待机模式下的功耗 32uA，130us 的快速切换和唤醒时间，世界领先的低功耗 RF24L01 特别适合采用钮扣电池供电的 2.4G 应用，与目前的蓝牙技术相比在提供更高速率的同时只需花更小的功耗。

2.3.2 RFL2401 通信流程解析

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

表 1 nRF24L01 工作模式

模式	PWR_UP	PRIM_RX	CE	FIFO 状态
接收	1	1	1	-
发射	1	0	1	数据已在发射堆栈里
发射	1	0	1→0	当 CE 有下降沿跳变时,数据已经发射
空闲 2	1	0	1	发射堆栈空
空闲 1	1	-	0	此时没有数据发射
掉电	0	-	-	-

Enhanced ShockBurst发送端工作流程:

RF24L01 发射数据时，首先将 RF24L01 配置为发射模式：接着把地址 TX_ADDR 和数据 TX_PLD 按照时序由 SPI 口写入RF24L01缓存区，TX_PLD 必须在CSN 为低时连续写入，而 TX_ADDR 在发射时写入一次即可，然后 CE 置为高电平并保持至少10μs，延迟 130μs 后发射数据;若自动应答开启，那么 RF24L01 在发射数据后立即进入接收模式，接收应答信号。如果收到应答，则认为此次通信成功，TX_DS 置高，同时 TX_PLD 从发送堆栈中清除;若未收到应答，则自动重新发射该数据(自动重发已开启)，若重发次数 (ARC_CNT) 达到上限，MAX_RT 置高，TX_PLD 会被清除;MAX_RT 或 TX_DS 置高时，使 IRQ 变低，以提示 MCU。最后发射成功时，若 CE 为低则 RF24L01 进入空闲模式 1;若发送堆栈中有数据且CE 为高，则进入下一次发射;若发送堆栈中无数据且 CE为高，则进入空闲模式 2。

Enhanced ShockBurst 接收端工作流程:

接收数据时，首先将 RF24L01 配置为接收模式，接着延

迟130μs进入接收状态等待数据的到来。当接收方检测到有效的地址和 CRC 时，就将数据包存储在接收堆栈中，同时中断标志位RX_DR置高，IRQ变低，以便通知MCU去取数据。若此时自动应答开启，接收方则同时进入发射状态回传应答信号。最后接收成功时，若CE变低，则 RF24L01 进入空闲模式 1。

控制 RF24L01 无线数据收发相关的配置字

SPI 口为同步串行通信接口，最大传输速率为 10 Mb/s，传输时先传送低位字节，再传送高位字节。但针对单个字节而言，要先送高位再送低位。与 SPI相关的指令共有8个，使用时这些控制指令由 RF24L01 的 MOSI 输入。相应的状态和数据信息是从 MISO 输出给 MCU。RF24L0l 所有的配置字都由配置寄存器定义，这些配置寄存器可通过 SPI 口访问。RF24L01 的配置寄存器共有 25 个，常用的配置寄存器如表 2 所示。

表 2 常用配置寄存器

地址(H)	寄存器名称	描述
00	CONFIG	可用来设置 nRF24L01 的工作模式
01	EN_AA Enhanced	用于接收通道的设置，使能接收通道的自动应答功能
02	EN_RXADDR	使能接收通道地址
03	SETUP_AW	设置地址宽度(适合所有通道)
04	SETUP_RETR	设置自动重发射
07	STATUS	状态寄存器
0A～0F	RX_ADDR_P0～P5	设置接收通道的地址
10	TX_ADDR	设置发射机地址
11～16	RX_PW_P0～P5	设置接收通道的数据长度

2.4 模块使用说明

(1) VCC(3.3V)脚接电压范围为 3V-3.6V 之间，不能在这个区间之外，超过 3.6V 将会烧毁模块。推荐电压 3.3V。当然，主要原因是 3.3V 的 LDO 芯片货源充足，也比较便宜，市场上面有台湾生产的批量价格 0.3RMB 一个的 LDO 芯片，本身功耗很低，压差也只有 50mV 不到，价格比 AMS1117-3.3 还便宜，非常不错。

(2) 除电源 VCC 和接地端，其余脚都可以直接和普通的 51 系列的 5V 单片机 IO 口直接相连,无需电平转换。这条规则当然对 3.3V 工作的单片机适用了。但是要注意本规则对于别的类型 5V 单片机不适用，由于有些类型单片机 IO 口驱动电流 > 10mA,所以需要串联限流电阻，一般说来,5V 工作的单片机和 3.3V 工作的无线模块之间通信，普通 IO 口线都加 1K-2K 的限流电阻.

(3) 硬件上面没有 SPI 的单片机也可以控制本模块，用普通单片机 IO 口模拟 SPI 不需要单片机 SPI 模块介入，只需添加代码模拟 SPI 时序即可。

(4) 17 脚、18 脚为接地脚,需要和母板的逻辑地连接起来

(5) 排针间距为 100mil,标准 DIP 插针，如果需要其他封装接口，比如密脚插针，或者其他形式的接口，可以联系我们定做。

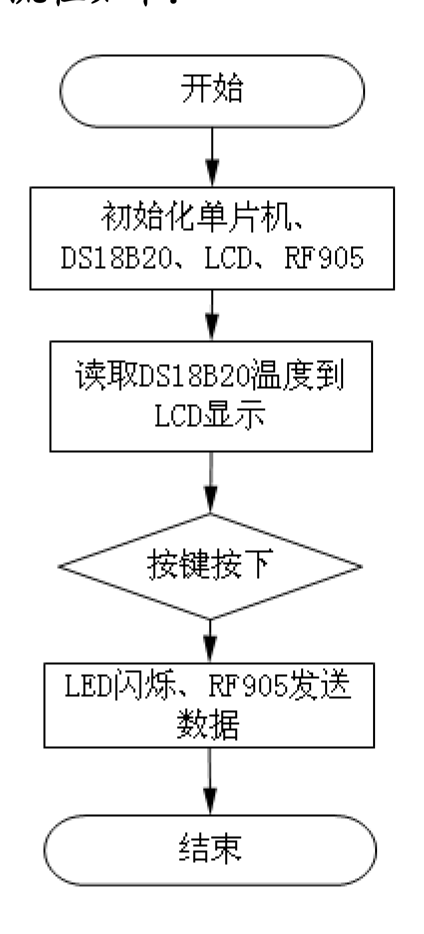
(6) 与 51 系列单片机 P0 口连接时候，需要加 10K 的上拉电阻,与其余口连接不需要。

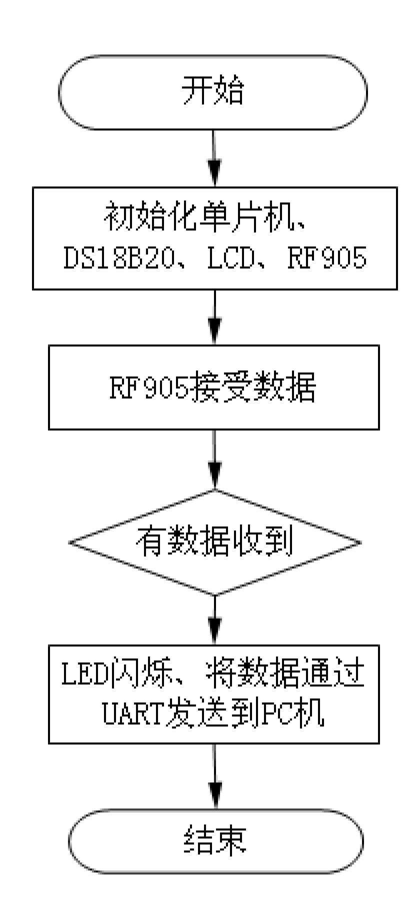
电话: 0571-81310832 13282105900 传真: 0571-86919079
网址: http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址: 浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

第三章 示例开发

3.1 开发代码流程

本套件在内部预置了开发代码，同时向客户提供源代码，其通信流程如下：





PC界面图：



3.2 部分代码示例

3.2.1 RF905 驱动程序

```
#define uchar unsigned char
#define uint  unsigned int

unsigned char TxAddress[4]={TX_ADDR_Byte0,TX_ADDR_Byte1,TX_ADDR_Byte2,TX_ADDR_Byte3};
```

```
RFConfig RxTxConf =
{
   10,
   RFConfig_Byte0,   RFConfig_Byte1,   RFConfig_Byte2,
   RFConfig_Byte3,   RFConfig_Byte4,
   RFConfig_Byte5,   RFConfig_Byte6,   RFConfig_Byte7,
   RFConfig_Byte8,   RFConfig_Byte9
};


static void Delay(uchar n)
{
   uint i;
   while(n--)
   for(i=0;i<80;i++);
}




/**********************************************************
**********************************/
//function InitIO();    初始化IO
/**********************************************************
**********************************/
void nRF905Init(void)
{
```

```
    /*SPI init*/
    SpiInit();
    //CSN=1;                  // Spi   disable

    //SCK=0;                  // Spi clock line init high
    DR=1;                  // Init DR for input
    AM=1;                  // Init AM for input
    PWR_UP=1;                 // nRF905 power on
    Delay(4);
    TRX_CE=0;                 // Set nRF905 in standby mode
    TX_EN=0;               // set radio in Rx mode

    //====================
    /*
    TX_EN_DDR   = 1;
    TRX_CE_DDR  = 1;
    PWR_UP_DDR  = 1;
    DR_DDR      = 0;
    CD_DDR      = 0;
    */
    //====================
}


/**********************************************************
**********************************/
//function WriteTxAddress();  写入目标地址
/**********************************************************
```

```
*********************************/
void WriteTxAddress(void)
{
   unsigned char i;
   unsigned char len= (RxTxConf.buf[2]>>4);
   CSN=0;                  // Spi enable for write a spi
command
   SpiWrite(WTA);          // Write address command
   //for (i=0;i<TX_AWF;i++)
   for (i=0;i<len;i++)
   {
      SpiWrite(TxAddress[i]);
   }
   CSN=1;                  // Spi disable
}


/*********************************************************
*********************************/
//function Config905();      //配置905寄存器
/*********************************************************
*********************************/
void Config905(void)
{
   uchar i;
   CSN=0;                  // Spi enable for write a spi
command
   SpiWrite(WC);           // Write config command写放配置命
```

令

```
   for (i=0;i<RxTxConf.n;i++) //  Write  configration  words
写放配置字
   {
      SpiWrite(RxTxConf.buf[i]);
   }
   CSN=1;                  // Disable Spi

   WriteTxAddress();
}


/**********************************************************
**********************************/
//function SetTxMode();       设置为发送模式
/**********************************************************
**********************************/
void SetTxMode(void)
{
   TRX_CE=0;
   TX_EN=1;
   Delay(1);               // delay for mode change(>=650us)
}

/**********************************************************
**********************************/
//function SetRxMode();       设置为接收模式
/**********************************************************
```

```
**********************************/
void SetRxMode(void)
{
   TX_EN=0;
   TRX_CE=1;
   Delay(1);               // delay for mode change(>=650us)

}




//发送数据包
/**********************************************************
**********************************/
void TxPacket(unsigned char *TxBuf)
{
   uchar i;

   CSN=0;                  //  Spi  enable  for  write  a  spi
command
   SpiWrite(WTP);          // Write payload command
   //for (i=0;i<TX_PW;i++)
   for (i=0;i<RxTxConf.buf[4];i++)
   {
      SpiWrite(TxBuf[i]);     // Write 32 bytes Tx data
   }
   CSN=1;                  // Spi disable
```

```
   TRX_CE=1;               // Set TRX_CE high,start Tx data
transmission
   while (DR!=1);
   TRX_CE=0;               // Set TRX_CE low
}

/*
**********************************************************
*********************************
*  读取数据包
*  返回值:     1-成功接收到数据     0-没有接收到数据
**********************************************************
*********************************
*/
unsigned char RxPacket(unsigned char *RxBuf)
{
   uchar i;
   if (DR)     //DR 高电平    有接收到数据
   {
      TRX_CE=0;               // Set nRF905 in standby mode


      CSN=0;                  // Spi enable for write a spi
command
      SpiWrite(RRP);          // Read payload command
      //for (i = 0 ;i < RX_PW ;i++)
      for (i = 0 ;i < RxTxConf.buf[3] ;i++)
```


电话: 0571-81310832 13282105900 传真: 0571-86919079
网址: http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN: wenming_hu2002@live.cn QQ: 4018749 908724356
地址: 浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼
416

```
      {
         RxBuf[i]=SpiRead();     // Read data and save to
buffer
      }
      CSN=1;                   // Disable spi

      TRX_CE=1;
        while(DR);//||AM);
      //SetRxMode();


      return 1;
   }
   else
   {
      return 0;
   }
}
```

3.3.2 LCD128*64 驱动程序

```
#define LCD_12864S_IO_INITIALIZTION()  \
LCD_12864s_CS1 = 1;\
LCD_12864s_RES = 0;\
_nop_();\
_nop_();\
_nop_();\
```

```
_nop_();\
LCD_12864s_RES = 1;\
_nop_();\
_nop_();\
_nop_();\
_nop_()

/******************宏函数*************************/
#define LCD_128164s_READ_DB() \
(LCD_12864s_DB  = 0xff)

#define LCD_12864s_WRITE_DB() \
(LCD_12864s_DB  = 0x00)

#ifdef ENABLE_LCD_12864s_BUSY
#define LCD_12864s_BUSY() \
LCD_128164s_READ_DB();\
LCD_12864s_DB  = 0xff;\
LCD_12864s_A0P = 0;\
LCD_12864s_WR  = 1;\
LCD_12864s_RD  = 0;\
LCD_12864s_CS1 = 0;\
while(LCD_12864s_DB_PIN>>7);\
(LCD_12864s_CS1 = 1)
#endif


#define     LCD_12864s_COMMEND_PROCESS(com_type,format,com)
```

```
((com_type&(~format))|(com&format))

#define LCD_12864s_DISPLAY_ON_OFF(factor)
LCD_12864s_COMMEND_PROCESS(0xae,0x01,factor)

#define LCD_12864s_START_LINE(factor)
LCD_12864s_COMMEND_PROCESS(0x40,0x3f,factor)

#define LCD_12864s_PAGE_ADDRESS(factor)
LCD_12864s_COMMEND_PROCESS(0xb0,0x0f,factor)

#define LCD_12864s_COLUMN_ADDRESS_H(factor)
LCD_12864s_COMMEND_PROCESS(0x10,0x0f,factor)
#define LCD_12864s_COLUMN_ADDRESS_L(factor)
LCD_12864s_COMMEND_PROCESS(0x00,0x0f,factor)

#define LCD_12864s_ADC_SELECT(factor)
LCD_12864s_COMMEND_PROCESS(0xa0,0x01,factor)

#define LCD_12864s_DISPLAY_NORMAL_REVERSE(factor)
LCD_12864s_COMMEND_PROCESS(0xa6,0x01,factor)

#define LCD_12864s_BIAS_SET(factor)
LCD_12864s_COMMEND_PROCESS(0xa2,0x01,factor)

#define LCD_12864s_RESET(factor)
LCD_12864s_COMMEND_PROCESS(0xe2,0x00,factor)
```

```
#define LCD_12864s_COMMON_OUTPUT_MODE(factor) LCD_12864s_COMMEND_PROCESS(0xc0,0x0f,factor)

#define LCD_12864s_POWER_CONTROLLER(factor) LCD_12864s_COMMEND_PROCESS(0x28,0x07,factor)

#define LCD_12864s_V5_REGULATOR(factor) LCD_12864s_COMMEND_PROCESS(0x20,0x07,factor)

#define LCD_12864s_ELECTRONIC_VOLUME_H(factor) LCD_12864s_COMMEND_PROCESS(0x81,0x00,factor)
#define LCD_12864s_ELECTRONIC_VOLUME_L(factor) LCD_12864s_COMMEND_PROCESS(0x00,0x3f,factor)


/*******************************************基 本 函 数*******************************************/



void Lcd12864sWdat(unsigned char dat)
{
#ifdef ENABLE_LCD_12864s_BUSY
  LCD_12864s_BUSY();
#endif
  LCD_12864s_WRITE_DB();
  LCD_12864s_A0P = 1;
  LCD_12864s_WR  = 0;
  LCD_12864s_RD  = 1;
```

```
  LCD_12864s_CS1 = 0;
  LCD_12864s_DB = dat;
  LCD_12864s_WR  = 1;
  LCD_12864s_WR  = 1;
  LCD_12864s_CS1 = 1;
}


void Lcd12864sWcom(unsigned char com)
{
#ifdef ENABLE_LCD_12864s_BUSY
  LCD_12864s_BUSY();
#endif
  LCD_12864s_WRITE_DB();
  LCD_12864s_A0P = 0;
  LCD_12864s_WR  = 0;
  LCD_12864s_RD  = 1;
  LCD_12864s_CS1 = 0;
  LCD_12864s_DB = com;
  LCD_12864s_WR  = 1;
  LCD_12864s_WR  = 1;
  LCD_12864s_CS1 = 1;
}

/****************************************** 扩 展 函 数 *******************************************/
/*初始化函数*/
void Lcd12864sInitializtion()
```

```
{
  LCD_12864S_IO_INITIALIZTION();
  Lcd12864sWcom(LCD_12864s_BIAS_SET(0x01));
  Lcd12864sWcom(LCD_12864s_ADC_SELECT(0x00));
  Lcd12864sWcom(LCD_12864s_COMMON_OUTPUT_MODE(0x00));
  Lcd12864sWcom(LCD_12864s_V5_REGULATOR(0x02));
  Lcd12864sWcom(LCD_12864s_ELECTRONIC_VOLUME_H(0x00));
  Lcd12864sWcom(LCD_12864s_ELECTRONIC_VOLUME_L(0x13));
  Lcd12864sWcom(LCD_12864s_POWER_CONTROLLER(0x07));
  Lcd12864sWcom(LCD_12864s_ADC_SELECT(0x01));
  Lcd12864sWcom(LCD_12864s_START_LINE(0x00));
  Lcd12864sWcom(LCD_12864s_DISPLAY_ON_OFF(0x01));
}


void Lcd12864sLocate(unsigned char page,unsigned char column)
{
  Lcd12864sWcom(LCD_12864s_PAGE_ADDRESS(page));

Lcd12864sWcom(LCD_12864s_COLUMN_ADDRESS_H((column+4)>>4));
  Lcd12864sWcom(LCD_12864s_COLUMN_ADDRESS_L(column+4));
}

void Lcd12864sCls()
{
  unsigned char temp1,temp2;
  for(temp1=0;temp1<8;temp1++)
  {
```

```
    Lcd12864sLocate(temp1,0x00);
    for(temp2=0;temp2<128;temp2++)
    {
      Lcd12864sWdat(0x00);
    }
    Lcd12864sWdat(0x00);
  }
}

/****************************************  常  用  函  数
*******************************************/

void Lcd12864sAscPrintf(unsigned char page,unsigned char
column,unsigned char const *p)
{
  unsigned char temp;

  Lcd12864sLocate(page,column);
  for (temp=0;temp<8;temp++)
  {
    Lcd12864sWdat(p[temp]);
  }
  Lcd12864sLocate(page+1,column);
  for (temp=8;temp<16;temp++)
  {
    Lcd12864sWdat(p[temp]);
  }
}
```

```
void  Lcd12864sBmpPrintf(unsigned  char  page,unsigned  char
column,unsigned char const *p)
{
  unsigned char width,height;
  unsigned int temp = 0x02;

  for(height=0;height<
p[1]/8+((p[1]%8!=0x00)&0x01);height++)
  {
    Lcd12864sLocate(page+height,column);
    for(width=0;width<p[0];width++)
    {
      Lcd12864sWdat(p[temp]);
      temp++;
    }
  }
}
```

3.3.3 DS18B20 读写程序

```
tempL=0; //设全局变量
unsigned char tempH=0;
unsigned char integer = 0;//测量到的温度的整数部分
unsigned char decimal1 = 0;//小数第一位
unsigned char decimal2 = 0;//小数第二位
unsigned char decimal = 0;//两位小数
unsigned char fg =0;
```

```
void delay_1ms(void)
{
   unsigned int i;

   for (i=1; i<(unsigned int)(xtal*142-2); ++i);   //xtal为
晶振频率，单位MHz
}

void delay_ms(unsigned int n)
{
   unsigned int i;

   for (i=0; i<n; ++i)
      delay_1ms();
}

void delay_n_us(unsigned char n)
{
   unsigned char i = 0;
   for (i=0; i<n-1; ++i)
   {
      _nop_();
   }
}
/*********************************************************
*********************/
```

```
void Delay_us(unsigned char i)//延时函数      num = 1   18.44
us 4   48us  8 65us    10  77.04us   100   663us
{
   while(i--);

}
/*********************************************************
********************/
void Init_DS18B20(void)//初始化ds1820
{
   unsigned char x=0;

   DQ = 1;     //DQ复位
   //Delay_us(8);  //稍做延时
   delay_n_us(10);
   DQ = 0;     //单片机将DQ拉低
   //Delay_us(80); //精确延时 大于 480us
   delay_n_us(480);
   DQ = 1;     //拉高总线
// Delay_us(8);      //65us
   delay_n_us(65);
   //x = DQ;       //稍做延时后 如果x=0则初始化成功 x=1则初始
化失败
   while (DQ == 0);
   Delay_us(20);
}
/*********************************************************
********************/
```

```
unsigned char ReadOneChar_18B20(void)//读一个字节
{
   unsigned char i=0;
   unsigned char dat = 0;
   for (i=8;i>0;i--)
   {
      DQ = 0; // 给脉冲信号
      dat>>=1;
      DQ = 1; // 给脉冲信号
      if(DQ)
      dat|=0x80;
      //Delay_us(4);
      delay_n_us(60);
   }
   return(dat);
}
/***********************************************************
*********************/
void WriteOneChar_18B20(unsigned char dat)//写一个字节
{                             //数据线从高电平拉至低电平，产
生写起始信号。15us之内将所需写的位送到数据线上
   unsigned char i=0;
   for (i=8; i>0; i--)
   {
      DQ = 0;
      DQ = dat&0x01;
      //Delay_us(5);
       delay_n_us(60);
```

```
      DQ = 1;
      dat>>=1;
   }
}
/************************************************************
*********************/
void Adjust_res_18B20(char res) ///res 分别等于 0x1f, 0x3f, 0x5f 温度读数分辨率分别对应
                          //              0.5, 0.25, 0.125
{
  Init_DS18B20();        //复位
  WriteOneChar_18B20(0xcc);  //跳过Rom
 WriteOneChar_18B20(0x4e);  //写暂存器
 WriteOneChar_18B20(0x02);  //写TH
 WriteOneChar_18B20(0x01);  //写TL
  //WriteOneChar_18B20(0x5f);  //写结构寄存器
  WriteOneChar_18B20(res);
  Init_DS18B20();       //复位
  WriteOneChar_18B20(0xcc); //跳过Rom
  WriteOneChar_18B20(0x48); //把暂存器内容写到EPRam中
}

/************************************************************
***********************/
void ReadTemperature(void)//读取温度
{
   //unsigned char a=0;
   //unsigned char b=0;
```

```
//unsigned int t=0;
//float tt=0;
Init_DS18B20();
WriteOneChar_18B20(0xCC); // 跳过读序号列号的操作
WriteOneChar_18B20(0x44); // 启动温度转换
delay_ms(800);
Init_DS18B20();
WriteOneChar_18B20(0xCC); //跳过读序号列号的操作
WriteOneChar_18B20(0xBE); //读取温度寄存器

tempL=ReadOneChar_18B20(); //读出温度的低位LSB
tempH=ReadOneChar_18B20(); //读出温度的高位MSB

if(tempH>0x7f)      //最高位为1时温度是负
{
   tempL=~tempL;         //补码转换，取反加一
   tempH=~tempH+1;
   fg=0;      //读取温度为负时fg=0
 }
integer = tempL/16+tempH*16;      //整数部分
decimal1 = (tempL&0x0f)*10/16; //小数第一位
decimal2 = (tempL&0x0f)*100/16%10;//小数第二位
decimal=decimal1*10+decimal2; //小数两位
```

（详细源代码请向本公司索取）

电话：0571-81310832 13282105900 传真：0571-86919079
网址：http://www.newmsg.com E-mail:wenming_hu2002@yahoo.com
MSN：wenming_hu2002@live.cn QQ：4018749 908724356
地址：浙江省杭州市下沙高教园杭州电子科技大学第二实验楼南楼416

第四章：关于我们

4.1 团队简介

NewMsg开发团队成立于2005 年1 月1 日，于2005 年8 月8 日在杭州注册 杭州源中通信技术有限公司。经过三年的发展，公司战略调整，于2008 年6 月16 日成立杭州威步科技有限公司。公司主要经营范围：

l 无线数传模块

l 无线音频传输、无线视频传输

l ARM9 开发平台

l 移动通信平台的开发与应用

l 有源RFID 系统、无源超高频读写器系统

我们团队由拥有博士和硕士文凭的专业人才组成，专注于电子相关领域，对光电子技术等当前最新电子技术进行实时跟踪，并开展了一些前期的相关工作。

欢迎电子行业同仁前来访问和洽谈项目合作，欢迎提出新产品需求。

4.2 联系方式

官方网站： http://www.NewMsg.com

官方技术讨论区：http://www.NewMsg.com/bbs

电话：0571-81310832 13282105900

传真：0571-86919079

Email:wenming_hu2002@yahoo.com

Skype: wenming_hu2002

MSN: wenming_hu2002@live.cn

QQ: 4018749

地址：浙江省杭州市江干区下沙高教园区杭州电子科技大学第二实验楼 4 层